(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2000－163456
（P2000－163456A）
(43)公開日　平成12年６月16日(2000.6.16)

(51)Int.Cl.⁷	識別記号	FI		テーマコード（参考）
Ｇ０６Ｆ　17/50		Ｇ０６Ｆ　15/60	６６４Ｐ	２Ｇ０３２
Ｇ０１Ｒ　31/28		11/22	３１０Ａ	５Ｂ０４６
Ｇ０６Ｆ　11/22	３１０		３４０Ａ	５Ｂ０４８
	３４０	Ｇ０１Ｒ　31/28	Ｆ	５Ｆ０６４
Ｈ０１Ｌ　21/82		Ｇ０６Ｆ　15/60	６６４Ｒ	９Ａ００１

審査請求　未請求　請求項の数３　ＯＬ　（全 13 頁）　最終頁に続く

(21)出願番号　特願平10－334119

(22)出願日　平成10年11月25日(1998.11.25)

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　中田　孝広
神奈川県秦野市堀山下１番地　株式会社日立製作所汎用コンピュータ事業部内
(72)発明者　鈴木　薫
神奈川県秦野市堀山下１番地　株式会社日立製作所汎用コンピュータ事業部内
(74)代理人　100080001
弁理士　筒井　大和

最終頁に続く

(54)【発明の名称】　論理検証方法

(57)【要約】
【課題】　操作性や観測性に優れた論理検証を、効率良く高速に行う。
【解決手段】　論理エミュレータ装置１０１に実装される、検証対象論理構造モデル１０５に当該検証対象論理構造モデル１０５のＩ／Ｏ環境等を提供する擬似論理モデル３０４を組み込み、テストベクトル３０３ａには、当該テストベクトル３０３ａの検証対象論理構造モデル１０５への入力による論理検証動作を制御するテスト制御プログラム３０５およびインタフェース情報３０６を組み込み、外部の汎用言語で記述されたＧＵＩインタフェースを持つ処理制御プログラム２０１から、インタフェース情報３０６を介して、擬似論理モデル３０４およびテスト制御プログラム３０５との間でテスト情報の授受を行うことで、ＧＵＩインタフェースにて論理検証の操作性と観測性を向上させ、高速な論理エミュレータ装置１０１による効率の良い大規模論理装置の一貫した論理検証を達成する。

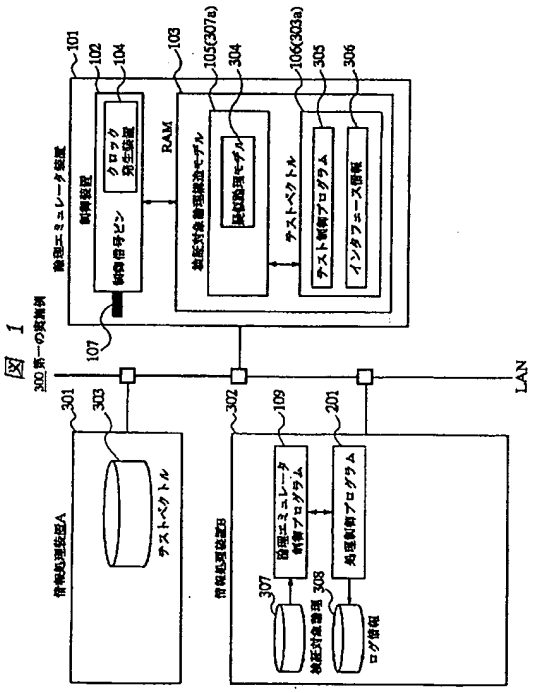


図１

1

【特許請求の範囲】

【請求項１】　検証対象の目的論理装置の論理構造モデルが実装された論理エミュレータまたは前記目的論理装置の実チップが実装された専用試験装置における論理検証に用いられるテストベクトルの中に、前記論理エミュレータまたは前記専用試験装置の外部に設けられた処理制御プログラムとの間で情報の授受を行うことで、前記テストベクトルの入力による前記論理検証を前記処理制御プログラムから制御可能にするテスト制御プログラムを実装することを特徴とする論理検証方法。

【請求項２】　請求項１記載の論理検証方法において、前記論理エミュレータでは、前記目的論理装置と等化な論理機能を実現するための第１の論理構造モデルと、前記第１の論理構造モデルの動作環境を提供する論理機能の少なくとも一部を実現するための第２の論理構造モデルとを実装し、前記目的論理装置の実行環境を実現するための外部ハードウェアとの接続を必要とすることなく、前記目的論理装置の論理検証を行うことを特徴とする論理検証方法。

【請求項３】　請求項１または２記載の論理検証方法において、

前記処理制御プログラムは、

前記テスト制御プログラムとの間における情報の授受を行う機能、

前記論理エミュレータを制御する論理エミュレータ制御プログラムとの間における情報の授受を行う機能、

前記目的論理装置の論理検証をソフトウェアにて行う論理シミュレータとの間における情報の授受を行う機能、

前記専用試験装置との間における情報の授受を行う機能、

前記テスト制御プログラム、前記論理エミュレータ制御プログラム、前記論理シミュレータ、前記専用試験装置の少なくとも一つとの間で授受される前記情報の可視化表示やユーザ入力の受け付けを行う汎用グラフィカル・ユーザ・インタフェース、

のうちの少なくとも一つを備えたことを特徴とする論理検証方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、論理検証技術に関し、特に、論理シミュレータや論理エミュレータを用いた論理回路装置の機能検証技術等に適用して有効な技術に関する。

【０００２】

【従来の技術】従来の論理エミュレーション装置（論理エミュレータ）を用いた論理エミュレーションは、論理検証対象のシステム装置や論理チップの一部分である論理モデルを動作記述言語で動作を記述し、設計された論理構造モデルを論理コンパイルして論理エミュレーション装置内のＲＡＭに実装し、論理モデル動作をエミュレーションするインサーキット方式で論理検証を行っていた。また、インサーキットではない装置や論理チップは、インサーキット論理とのインタフェースを確保するために論理エミュレーション装置の外部接続ピンに互いの外部信号線を接続し、信号値レベルで動作の同期をとっている。また、論理動作の観測やテストベクトルを個別に指定する場合には、論理エミュレーション装置内の観測可能な内部信号や外部ピンに対して論理値を与えたり、あらかじめ指定された信号線に対して信号線単位での観測を行い、論理動作の確認を行っていた。

2

【０００３】このような論理エミュレーション技術に関しては、特開平０２－２４５８３１号公報に示されたような方法等が知られている。すなわち、検証対象の論理機能を外部からプログラム可能なゲートアレイ上にマッピングすることにより、目的の論理機能をハードウェア的に実現して、高速な実行および論理検証を可能にしようとするものである。

【０００４】

【発明が解決しようとする課題】従来のインサーキット方式は、近年の論理規模の増大や論理の複雑さに対して部分論理での論理エミュレーションのためシステムテストのような論理品質を確保するには充分とは言えず、論理チップ全体での論理エミュレーションによるシステム論理検証が必要になってきている。論理チップ全体での論理検証では、論理規模の増大や論理の複雑化も進み、従来の信号線レベルでの論理値を与えたり、信号線の値を観測したりする論理エミュレーションでは信号線全てに論理値を設定しなければならないため、効率の良い論理検証を行うことが難しい。観測性や操作性を向上させるには、マンマシンインタフェースを充実させ、テスト容易性を向上させる必要がある。

【０００５】また、論理チップ全体での論理シミュレーションは、論理シミュレータではソフトウェアで論理モデルを作成するため、ハードウェア動作に比べてソフトウェア動作のため論理検証に莫大な時間を要し、製品を短期開発し、早期出荷しなければならない今日では適用が難しく、解決しなければならない課題の一つである。

【０００６】また、検証対象の論理が周辺装置などの論理の場合には、論理単体での論理検証動作を確認出来ない場合もあり、検証レベルの異なるアーキテクチャ論理シミュレータ等を併用して検証対象論理の論理検証を行ったり、専用の論理シミュレータを用いたりして論理検証を行うことが必要になる。したがって、効率の良い論理シミュレータを選択し、システムテストレベルで併用出来る論理エミュレーション方式の構築が重要な課題である。

【０００７】また、論理エミュレーションが終了した段階で実チップが製造され、チップ単体または装置全体での論理品質検査を実施する時、それまでの論理検証環境を継続して使用できれば論理不良が発見された場合に論

理シミュレーションや論理エミュレーションへのフィードバックが容易であり、検証環境の構築ということを考えれば一貫した論理検証環境の構築が必要である。

【０００８】本発明の目的は、論理チップ全体での論理エミュレーションによるシステム論理検証を短時間に効率よく行うことが可能な論理検証技術を提供することにある。

【０００９】本発明の他の目的は、マンマシンインタフェースを充実させ、論理エミュレーションにおける内部状態の観測性や操作性、さらにはテスト容易性を向上させることが可能な論理検証技術を提供することにある。

【００１０】本発明の他の目的は、論理モデルから実チップに至るまでの一貫した論理検証を実現することが可能な論理検証技術を提供することにある。

【００１１】本発明の他の目的は、論理検証の環境構築に要する工数や期間を短縮して、論理検証工程におけるコスト削減を実現することが可能な論理検証技術を提供することにある。

【００１２】

【課題を解決するための手段】本発明は、目的論理装置の論理検証を目的として、論理エミュレータ上に目的論理装置の論理構造モデルと、論理構造モデルと共に動作することで当該論理構造モデルの実行環境を提供する擬似論理モデルを実装し、論理エミュレータ上で実行されるテストベクトルには、外部の処理制御プログラムとの間で情報の授受を行うことで論理検証動作を制御するテスト制御プログラムを実装するものである。

【００１３】擬似論理モデルとプログラムと論理エミュレータを制御する処理制御プログラムは、たとえばネットワークを介して論理エミュレータに接続された情報処理装置上に実装され、論理エミュレータと当該処理制御プログラムの間で情報通信を行う手段と、論理エミュレータを非同期に動作させる手段と、各種情報の可視化表示や情報の入力環境等を提供するグラフィカルユーザーインタフェースと、論理エミュレータ以外の論理シミュレータ等の検証手法の異なる論理検証プログラムを当該処理制御プログラムに接続する接続プラグインタフェースと、論理エミュレータ上で動作するテスト制御プログラムやテストベクトルとのインタフェースとを有しており、目的論理装置の論理検証の操作性と観測性を向上させ効率よく高速に論理検証を可能とする。

【００１４】また、目的論理装置の実チップが実装される専用試験装置において、実チップに入力されるテストベクトル内に、外部の処理制御プログラムとの間で情報の授受を行うことで論理検証動作を制御するテスト制御プログラムを実装するものである。この場合、処理制御プログラムは、上述の構成の他に、任意の接続インタフェースを介して専用試験装置が接続される情報処理装置に実装され、前記接続インタフェースを制御するデバイスドライバとの情報の授受を行うドライバ制御プログラムを有し、実チップの論理検証における操作性や観測性を向上させ効率良く高速に論理検証を行う。また、テストベクトルや、テスト制御プログラム、処理制御プログラムは、論理エミュレータの場合と共通のものを用いることができる。

【００１５】また、テストベクトル内のテスト制御プログラムと擬似論理モデルとのインタフェースを持つ処理制御プログラムにおいて、論理エミュレータに実装した目的論理装置の論理モデルの内部信号値の観測や制御を行い、処理状態をＩ／Ｏリクエストとして制御し、テスト制御プログラムとＩ／Ｏリクエストを処理制御プログラムが送受信することで、論理エミュレータを制御しながら目的論理装置の論理検証を行うことができる。

【００１６】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照しながら詳細に説明する。

【００１７】図１は本発明の第一の実施の形態である論理検証方法が実施される情報処理システムの構成の一例を示す概念図であり、図２は、本発明の論理検証方法を実現するためのソフトウェアの構成の一例を例示した概念図である。また、図８は、本実施の形態の論理検証方法の参考技術の論理エミュレーションシステムの構成の一例を示す概念図である。

【００１８】先ず、図８を用いて本実施の形態の説明の前に、本実施の形態の参考技術の論理エミュレーションシステムの概要を説明する。

【００１９】図８の論理エミュレーションシステム１００は、ＬＡＮ接続された論理エミュレータ装置１０１と、同じくＬＡＮ接続された情報処理装置１０８上で動作する論理エミュレータ制御プログラム１０９で構成される。論理エミュレータ装置１０１は、制御装置１０２と読み書き可能なＲＡＭ１０３で構成され、制御装置１０２には論理エミュレータ装置１０１に供給される動作クロックを発生させるクロック発生装置１０４と、論理エミュレーション制御用の制御信号ピン１０７などで構成される。なお、本発明に関する構成要素のみについて特化して説明を行う。

【００２０】制御信号ピン１０７は、エミュレーション状態を論理エミュレータ制御プログラム１０９に報告したり、動作指示を受け取るための外部ピンである。ＲＡＭ１０３は、検証対象論理構造モデル格納エリア１０５に後述の検証対象論理構造モデル１１４ａを実装したり、テストベクトル格納エリア１０６にテストベクトルを実装するために用いる。論理エミュレータ制御プログラム１０９は、論理エミュレータ装置１０１および論理エミュレータ制御プログラム１０９を制御するための制御部１１０と、ＲＡＭアクセス部１１２と、動作記述言語で記述された検証対象論理１１４を論理コンパイルする論理コンパイラ１１１と、論理エミュレーションシステム全体をグラフィカルに表示、制御するグラフィカル

ユーザーインタフェース（ＧＵＩ）制御部１１３で構成される。論理コンパイラ１１１は、検証対象論理１１４を入力とし、論理コンパイルして検証対象論理構造モデル１１４ａとして論理エミュレータ装置１０１内のＲＡＭ１０３の検証対象論理構造モデル格納エリア１０５に実装する。

【００２１】次に論理エミュレーションを実際に行う手順について説明する。前述した検証対象論理１１４を論理コンパイルして検証対象論理構造モデル１１４ａとしてＲＡＭ１０３に実装した後、テストベクトルをＲＡＭアクセス部１１２によりテストベクトル格納エリア１０６に実装し、制御部１１０によって制御信号ピン１０７に対して動作条件情報を送り、クロック発生装置１０４によって実際にクロックが供給され、論理エミュレータ装置１０１を動作させ論理エミュレーションを行う。次に一定サイクル論理エミュレーションを行った後、制御装置１０２は、制御信号ピン１０７にエミュレーション動作状態情報をセットし、この情報を制御部１１０が監視し、その状態情報に合わせて引き続きクロックを供給したり、中断してＲＡＭ１０３の状態を観測したりする。これらの処理を繰り返し行うことで論理検証を行っている。

【００２２】次に図２に本発明の各実施の形態の論理検証方法を実現するための処理制御プログラムの一例を示す。処理制御プログラム２０１は、ネットワーク上に起動された論理エミュレータ制御プログラム１０９との間でプロセス間通信を行うプロセス間通信制御部２０３と、論理エミュレータ制御部２０４と、テストベクトル制御部２０５と、論理エミュレータ装置１０１との間で送受信されるデータの変換処理を行うデータ変換部２０６と、論理エミュレータ装置１０１以外の検証装置および検証システムを処理制御プログラム２０１に接続するための接続プラグインタフェース２０７と、本発明の論理検証方式を総合的にコントロールするグラフィカルユーザーインタフェース（ＧＵＩ）制御部２０２、で構成される。なお、詳細な各処理部の説明は実施の形態の説明で行う。また、図２には、簡単のため、プロセス間通信制御部２０３～接続プラグインタフェース２０７の構成要素をすべて含む構成が例示されているが、これらのうちの必要な構成要素のみを含む構成も本発明に含まれる。

【００２３】次に図１を用いて本発明の第一の実施の形態である論理検証方法が実施される論理エミュレーションシステムを説明する。図１に示した第一の実施の形態の論理エミュレーションシステム３００において、検証対象論理を機能レベルアーキテクチャを実現した論理として論理コンパイルして論理エミュレータ装置１０１のＲＡＭ１０３にセットし、論理検証を行う手法を説明する。

【００２４】なお、本実施の形態の論理エミュレーションシステム３００では、論理エミュレータ装置１０１は、インサーキット方式の論理検証において、後述の擬似論理モデル３０４を論理エミュレータ装置１０１内に実装することにより、検証対象の論理構造モデルのみを実装した状態でのベクトル検証を可能にするものであり、検証対象の論理モデル（と等化な動作を行う実チップ）の実際の動作環境を提供するために周辺のハードウェア回路を論理エミュレータ装置１０１に接続する必要はない。

【００２５】たとえば検証対象がマイクロプロセッサである場合、その全体の論理機能の検証には、実際のシステムの構築に用いられる周辺回路との入出力環境を実現する必要があるため、図８に例示されたような参考技術の論理エミュレーションシステム１００では、あらかじめ、周辺回路のハードウェア環境を準備しておき、マイクロプロセッサの動作をエミュレートする論理エミュレータ装置１０１に対して配線接続する必要がある。これに対して、本実施の形態の論理エミュレーションシステム３００の場合には、後述の擬似論理モデル３０４が、目的の検証対象論理構造モデル３０７ａの動作に必要なＩ／Ｏインタフェースを実現するので、実際の周辺回路等への接続（インサーキット接続）は不要であり、論理エミュレータ装置１０１の単体で検証動作が可能である。

【００２６】本実施の形態では、テストベクトル３０３は、ＬＡＮに接続された情報処理装置３０１（Ａ）に格納されている。また、ＬＡＮに接続された情報処理装置３０２（Ｂ）には、論理エミュレータ制御プログラム１０９および処理制御プログラム２０１が実装されて実行される。情報処理装置３０１（Ａ）や情報処理装置３０２（Ｂ）は、たとえばＵＮＩＸ等のＯＳで動作するワークステーションやパーソナルコンピュータ等で構成される。

【００２７】先ず、前準備として検証対象論理３０７とテストベクトル３０３を準備する。前記の検証対象論理３０７には、処理制御プログラム２０１とのインタフェース機能を有する擬似論理モデル３０４を組み込み、論理コンパイラ１１１で論理コンパイルして検証対象論理構造モデル３０７ａとしておく。擬似論理モデル３０４の機能としては、論理動作中の内部信号線の論理値を観測したり監視したりする機能と、検証対象論理３０７（検証対象論理構造モデル３０７ａ）の動作をコントロールするシステムコントロール機能と処理制御プログラム２０１とのインタフェース機能を有する。また、テストベクトル３０３には、処理制御プログラム２０１とのインタフェース機能を持つテスト制御プログラム３０５と、処理制御プログラム２０１とのインタフェースに使用するインタフェース情報３０６を組み込んでおく。

【００２８】次に、論理エミュレータ制御プログラム１０９を起動し、準備した検証対象論理３０７のコンパイ

ル結果を、ＲＡＭアクセス部１１２を用いて、検証対象論理構造モデル３０７ａとしてＲＡＭ１０３内の検証対象論理構造モデル格納エリア１０５にセットする。

【００２９】次に処理制御プログラム２０１を起動し、準備しておいたテストベクトル３０３をデータ変換部２０６でＲＡＭ１０３にセット可能な形式のテストベクトル３０３ａに変換し、論理エミュレータ制御部２０４内のＲＡＭアクセス部２１０経由でＲＡＭ１０３内のテストベクトル格納エリア１０６にセットする。

【００３０】図８に示した参考技術の論理エミュレーションシステム１００と異なる点は、擬似論理モデル３０４とテストベクトル３０３ａ内のテスト制御プログラム３０５およびインタフェース情報３０６、および論理エミュレータ制御プログラム１０９と独立な処理制御プログラム２０１が設けられていることである。

【００３１】また、本実施の形態の論理エミュレータ制御プログラム１０９は、Ｔｃｌ／Ｔｋインタフェースを有するため、処理制御プログラム２０１をＴｃｌ／Ｔｋで作成することにより、参考技術である図８の論理エミュレータ制御プログラム１０９の機能をそのまま使用できると共に論理エミュレータ制御プログラム１０９の拡張機能の一貫として容易に接続可能である。つまり、処理制御プログラム２０１は、論理エミュレータ装置１０１を外部から操作可能にする拡張プログラムと言える。

【００３２】なお、Ｔｃｌ／Ｔｋとは、カリフォルニア大学バークレー校で開発されたグラフィカルインタフェース（ＧＵＩ）を有する汎用スクリプト言語であり、オープンシェルスクリプトとして多くの大学や研究所の他、大小様々な企業で利用されている。

【００３３】また、本発明で採用したＴｃｌ／Ｔｋインタフェースは、本実施例の論理エミュレータ装置がＴｃｌ／Ｔｋインタフェースを有していたためであり、本発明を実施する際、マンマシンインタフェースを提供するＧＵＩ機能はＸウインドウシステムやＭｏｔｉｆ等で実現することが可能である。更に言い換えれば、ＧＵＩインタフェースを持ったシステムであれば、その機能やシステムを限定するものではない。

【００３４】次に実際に論理エミュレーションを行う手順に沿って説明する。テストベクトル３０３ａがセットされた後、設定情報制御部２１５により詳細な論理エミュレーション動作の設定やテスト制御プログラム３０５への設定、インタフェース情報３０６の詳細設定、動作周波数など論理エミュレーション動作に必要な各種設定を行う。これらの設定は、ＧＵＩ制御部１１３やＧＵＩ制御部２０２によって制御されている。各種設定が完了した後、論理エミュレータ制御部２０４内の動作制御部２０９が擬似論理モデル３０４に対してリセット要求を出し、検証対象論理構造モデル３０７ａの状態を動作可能な状態にする。この後、制御信号ピン１０７に対して動作クロック数を与え、制御装置１０２はクロック発生装置１０４により、指定されたクロック数分だけクロックが論理エミュレータ装置１０１に供給され実際に動作する。指定された一定クロック数分だけ動作しているとき、テスト制御プログラム３０５はインタフェース情報３０６に対して、内部動作状態やテスト動作状態などをセットする。また、指定されたクロック数に到達することなく、エミュレーションを中断したり、処理制御プログラム２０１に対してメッセージ等の出力要求がテスト制御プログラム３０５で発生した場合には、その状態情報をインタフェース情報３０６にセットする。

【００３５】擬似論理モデル３０４は、インタフェース情報３０６を監視しているため、これらの要求を検出すると、制御装置１０２の制御信号ピン１０７に対して、トリガーイベントを発生させ制御装置１０２に中断報告を行う。このように擬似論理モデル３０４は、エミュレーション動作を中断させる機能を有し、処理制御プログラム２０１と直接対話型で情報のやりとりが可能である。つまり、論理内部の内部信号の値を監視したりすることで、エミュレーションを中断し、その情報を処理制御プログラム２０１に伝達可能である。

【００３６】次に、処理制御プログラム２０１のトリガーイベント制御部２０８が、制御信号ピン１０７の監視をしているため、トリガーイベントを検出すると中断情報の格納されているインタフェース情報３０６をＲＡＭアクセス部２１０経由で取得し、取得した情報は、表示するためにデータ変換部２０６で変換し、表示部２１２に引き渡し表示を行う。中断情報に対して、テストベクトル制御部２０５で取得情報の解析を行い、テスト制御プログラム３０５が入力またはコマンド等を要求している場合には、入力制御部２１３により、オペレーターのキー入力やＧＵＩウインドウ操作を受付け、入力された情報をインタフェース情報３０６にセットする。セットした後、クロックを供給すれば論理エミュレーションは継続される。

【００３７】なお、これらのエミュレーション結果や動作中の状態を保存する場合には、ログデータ採取制御部２１６により、表示された情報や入力された情報、およびアクセス可能なＲＡＭ情報などを取得し、ログ情報３０８として保存可能である。また、処理制御部２１１とウインドウ制御部２１４は、一貫して処理制御プログラム２０１のコントロールを行い、各種操作をサポートする。

【００３８】以上のようにＧＵＩ制御部２０２の各種機能とテストベクトル制御部２０５を用いてＲＡＭ１０３内のテスト制御プログラム３０５と対話式に論理エミュレーションを行うことで論理エミュレータ装置１０１内の専用の擬似論理モデル３０４との対話手段を確保することが可能となり、論理エミュレーション動作において、インタフェース情報３０６のみの操作で論理エミュレーションの制御が可能であり、操作性や観測性を向上

させ、効率の良い論理検証が可能である。また、これらの手法は、論理エミュレータ制御プログラム１０９を間接的に制御しているため、図８に例示された参考技術を包含し、その上で外部から論理エミュレータ装置１０１を制御可能である。更に、テスト制御プログラム３０５で論理エミュレーション動作をコントロール可能であるため、論理エミュレーションを中断させることを最小限にし、論理エミュレータ装置１０１の高速性を最大限に利用できるため、高速な論理エミュレーションが可能である。

【００３９】次に第一の実施の形態の論理エミュレーションにおける具体的な処理フローの一例を図３に示す。図３では、論理エミュレーション動作途中でのテスト制御プログラム３０５がメッセージ出力要求とコマンド入力要求を発生したときの本実施の形態の構成部分の作用についてのみの処理フローを説明する。先ず、図３での処理ステージは、オペレータ４０１、処理制御プログラム２０１、擬似論理モデル３０４、テスト制御プログラム３０５のステージに区分し、これら各ステージの連携動作を処理フローとして示す。

【００４０】前述の検証対象論理３０７とテストベクトル３０３が、それぞれ検証対象論理構造モデル３０７ａおよびテストベクトル３０３ａとしてＲＡＭ１０３にセットされ、クロックを供給されればエミュレーションが開始可能な状態である場合において、オペレータは動作開始指示を行う（ステップ４０２）。このステップ４０２の動作開始指示には、動作クロック数（動作サイクル数）が指示されている。次に、処理制御プログラム２０１はエミュレーション開始を実際に行い（ステップ４０３）、論理エミュレータ装置１０１は論理エミュレーションを開始する。論理エミュレーションが開始されると擬似論理モデル３０４とテスト制御プログラム３０５が動作を開始する（ステップ４０４）。この時、擬似論理モデル３０４はインタフェース情報３０６の監視を常に行っている（ステップ４０５）。そして、テスト制御プログラム３０５が動作中にメッセージをオペレータに報告するため、メッセージ出力要求が発生し、更にそのメッセージに対する応答コマンドの入力要求が発生した場合（ステップ４０６）、テスト制御プログラム３０５はインタフェース情報３０６に出力メッセージと擬似論理モデル３０４が検出可能なイベント情報をセットする（ステップ４０７）。次に擬似論理モデル３０４はインタフェース情報３０６の監視によりイベントの検出を行い（ステップ４０８）、制御信号ピン１０７に対してトリガーイベントを発生させる（ステップ４０９）。トリガーイベントが発生すると制御装置１０２はエミュレーションを中断し（ステップ４１０）、処理制御プログラム２０１は中断状態からトリガーイベントによる中断であることを認識し（ステップ４１１）、インタフェース情報３０６をＲＡＭ読み出しにより取得する（ステップ４１２）。取得した情報はデータ変換部２０６を通してデータ変換し（ステップ４１３）、テスト制御プログラム３０５からの出力メッセージとして表示部２１２のウインドウに表示する（ステップ４１４）。通常は、メッセージの出力のみであれば、表示後エミュレーションを再開するが、この実施の形態の場合には入力コマンド要求も同時に発生しているため、コマンドの入力要求をオペレータ４０１に対して行い（ステップ４１５）、処理制御プログラム２０１は入力待ち状態になる（ステップ４１７）。オペレータ４０１は、キー入力を行い（ステップ４１６）、テスト制御プログラム３０５に対するコマンドを入力する。次にキー入力されたコマンドを取得し（ステップ４１８）、データ変換部２０６でデータ変換を行い（ステップ４１３）、インタフェース情報３０６にセットすべく、ＲＡＭ書き込みを行う（ステップ４１９）。次に、エミュレーション動作を再開し（ステップ４２０）、擬似論理モデル３０４とテスト制御プログラム３０５は動作を再開する（ステップ４２１）。再開後は、インタフェース情報３０６から入力コマンドを取得し（ステップ４２２）、そのコマンドに合わせた動作を行う（ステップ４２３）。

【００４１】このように、オペレータ４０１とテスト制御プログラム３０５の間でインタフェース情報を相互に転送し、対話的に論理エミュレーションを行うことで、オペレータは外部から論理エミュレータ装置１０１を制御し、テスト制御プログラム３０５は内部から論理エミュレータ装置１０１を制御することで効率の良い論理検証が可能である。なお、テスト制御プログラム３０５と同様にインタフェース情報３０６を監視する擬似論理モデル３０４も論理エミュレータ装置１０１を内部から制御可能であり、特に論理モデルであるため、内部信号などの観測性に優れ、擬似論理モデル３０４はハードウェア的に、テスト制御プログラム３０５はソフトウェア的に、論理エミュレータ装置１０１を制御可能であり、効率の良い論理エミュレーションが可能である。

【００４２】すなわち、本実施の形態によれば、テストベクトル格納エリア１０６のテストベクトル３０３ａ内に設けられたテスト制御プログラム３０５や、擬似論理モデル３０４によりテストベクトル３０３ａとのインタフェースを確保し、テストベクトル３０３ａからのＩ／Ｏリクエストを処理制御プログラム２０１が送受信し、Ｔｃｌ／Ｔｋ等の汎用のＧＵＩ機能を用いて操作性や観測性を向上させることにより、論理エミュレーションでのシステム論理検証を効率良く高速に行うことが可能である。

【００４３】次に本発明の第二の実施の形態である論理検証方法が実施される論理エミュレーションシステム５００を図４を参照して説明する。図４の論理エミュレーションシステム５００では論理エミュレータ装置１０１の代わりに専用情報処理装置などを接続し、検証論理チ

ップ５０８の論理検証を行う手法を説明する。構成としては、ＬＡＮ接続された情報処理装置３０１（Ａ）は第一の実施の形態と同じでテストベクトル３０３も同じものである。また、情報処理装置５０１（Ｃ）は、この第二の実施の形態では一例として、たとえば汎用のパーソナルコンピュータ用ＯＳであるＷｉｎｄｏｗｓ系ＯＳで動作するパーソナルコンピュータとして説明する。情報処理装置５０１（Ｃ）は、Ｔｃｌ／Ｔｋインタプリタ５０３と、処理制御プログラム２０１と、ドライバ制御プログラム６０１と、デバイスドライバ５０５で構成される。

【００４４】そして、情報処理装置ＣにパラレルＩ／Ｏインタフェース５０６で接続される専用情報処理装置５０２は、制御装置５０７と、検証論理チップ５０８と、ＲＡＭ５０９で構成される。ここでの検証論理チップ５０８は、第一の実施の形態での検証対象論理構造モデル３０７ａ（検証対象論理３０７）を実際のチップとして製造したものである。そして、ＲＡＭ５０９のテストベクトル格納エリア１０６には、第一の実施の形態で使用したテストベクトル３０３ａ（テストベクトル３０３）をそのままセットする。

【００４５】デバイスドライバ５０５は、Ｗｉｎｄｏｗｓ系ＯＳのデバイスドライバでパラレルＩ／Ｏインタフェース５０６を制御可能なデバイスドライバである。そして、このデバイスドライバ５０５を制御するためのドライバ制御プログラム６０１は、処理制御プログラム２０１の接続プラグインタフェース２０７とのインタフェースを持つことにより、処理制御プログラム２０１とデバイスドライバ５０５とを接続し、処理制御プログラム２０１がシステム全体を制御可能とする。そして、デバイスドライバ経由で制御装置５０７を制御することで専用情報処理装置５０２を制御する。

【００４６】また、Ｔｃｌ／Ｔｋインタプリタ５０３は、Ｗｉｎｄｏｗｓ系ＯＳ版のＴｃｌ／Ｔｋのインタプリタであり、処理制御プログラム２０１は、第一の実施の形態のＵＮＩＸ系ＯＳ版のものがそのままＷｉｎｄｏｗｓ系ＯＳ版でも動作可能である。

【００４７】したがって、この第二の実施の形態の検証目的は、実機に搭載する論理チップが製造された時点で、論理エミュレーションで用いたテストベクトル３０３と処理制御プログラム２０１を用いて、論理チップの論理品質を検証することである。つまり、実機システム検証の前に論理チップ単体の論理品質検証を行うことである。

【００４８】次に、ドライバ制御プログラム６０１の概要を図５に示す。ドライバ制御プログラム６０１は、制御部６０２と、処理制御プログラムインタフェース部６０３と、デバイスドライバ制御部６０４と、ログデータ採取制御部６０５で構成される。制御部６０２は、ドライバ制御プログラム６０１の全体の制御を行い、処理制御プログラムインタフェース部６０３は、処理制御プログラム２０１からのＲＡＭアクセスや専用情報処理装置５０２の制御指示を制御する。デバイスドライバ制御部６０４は、デバイスドライバ５０５とのインタフェースを持ち、デバイスドライバ経由でパラレルＩ／Ｏインタフェース５０６を制御する。ログデータ採取制御部６０５は、ドライバ制御プログラム６０１で取得可能なログデータを採取し、ログ情報５０４として出力あるいは保存する機能を持つ。

【００４９】次に第二の実施の形態の具体的な処理フローを図６に示し説明する。図６では、専用情報処理装置５０２が動作途中でのテスト制御プログラム３０５がメッセージ出力要求とコマンド入力要求を発生したときの本発明の部分に着目して処理フローを説明する。先ず、図６での処理ステージは、オペレータ７０１、処理制御プログラム２０１、ドライバ制御プログラム６０１、テスト制御プログラム３０５のステージに区分されており、全体処理フローを表している。

【００５０】前述の検証論理チップ５０８が専用情報処理装置５０２にセットされ、テストベクトル３０３ａがＲＡＭ５０９のテストベクトル格納エリア１０６にセットされ、動作開始指示待ち状態である場合において、オペレータ７０１は検証開始指示を行う（ステップ７０２）。次に処理制御プログラム２０１は、ドライバ制御プログラム６０１に対して動作開始指示を行い（ステップ７０３）、ドライバ制御プログラム６０１は、専用情報処理装置５０２を作動させる（ステップ７０４）。装置が作動するとテスト制御プログラム３０５は動作を開始する（ステップ７０５）。また、処理制御プログラム２０１は、ステップ７０３を行った後、インタフェース情報の監視を開始する（ステップ７０６）。インタフェース情報３０６の監視では、ＲＡＭ５０９のポーリングアクセスのため、ドライバ制御プログラム６０１は、常にＲＡＭアクセスを行い、ＲＡＭ読み出しを行っている（ステップ７０７）。これらの状態であるとき、テスト制御プログラム３０５は、メッセージの出力要求とコマンド入力要求が発生すると（ステップ７０８）、メッセージとイベントをインタフェース情報にセットする（ステップ７０９）。次にインタフェース情報３０６の監視を行っている処理制御プログラム２０１が、このイベントを要求として検出し（ステップ７１０）、インタフェース情報３０６を取得する（ステップ７１１）。取得したデータはデータ変換され（ステップ７１２）、メッセージとしてウインドウに表示される（ステップ７１３）。表示後は、オペレータ７０１がコマンド入力要求（ステップ７１４）に対してキー入力やウインドウ操作（ステップ７１５）を行う。

【００５１】次に入力待ち状態の処理制御プログラム２０１は（ステップ７１６）、入力されたキー入力を取得し（ステップ７１７）、データ変換を行い（ステップ７

１２）、インタフェース情報３０６としてＲＡＭ５０９に書き込むことにより（ステップ７１８）、テスト制御プログラム３０５に転送する（ステップ７１９）。転送後は処理制御プログラム２０１は再度、インタフェース情報３０６の監視状態に入る（ステップ７０６）。テスト制御プログラム３０５は、前記ステップ７０８の要求を出した後も動作中であり（ステップ７２０）、転送されたインタフェース情報３０６から入力イベントを検出すると（ステップ７２１）、インタフェース情報３０６から入力コマンドを取得し（ステップ７２２）、取得したコマンドに対する動作を行う（ステップ７２３）。このような一連の動作を行い、対話的に検証論理チップ５０８の論理検証を行う。

【００５２】また、第一の実施の形態と共通したテストベクトル３０３と処理制御プログラム２０１をそのまま使用可能であり、検証対象論理構造モデル１０５を用いた論理エミュレーションと、検証論理チップ５０８等の実チップの単体検証を同じ環境で行うことが可能であり、論理モデルから実チップに至るまでの一貫した論理検証が可能となる。

【００５３】すなわち、論理エミュレータ装置１０１を用いた論理エミュレーションによる論理検証工程と、検証論理チップ５０８等の実チップによる論理検証工程に共通のソフトウェアやテストベクトルを使用できることで、各工程別にソフトウェアやテストベクトルを用意する等の重複した労力を軽減でき、効率の良い論理検証が可能となる。

【００５４】次に本発明の第三の実施の形態である論理検証方法が実施される論理エミュレーションシステム８００を図７に示す。図７の論理エミュレーションシステム８００では図１の第一の実施の形態に論理シミュレータを加え、論理検証手法と論理検証レベルの異なる論理エミュレータと論理シミュレータが混在する論理検証方式の場合が例示されている。

【００５５】この第三の実施の形態としては、具体的には、たとえば論理シミュレータ８０２でマイクロプロセッサ等の論理動作を行わせ、論理エミュレータ装置１０１では、このマイクロプロセッサを含むシステムで使用されるＩ／Ｏ制御デバイス等の論理を実装することで、双方の実チップを使用したシステムを組み上げる前に、両者の連携した動作における双方の論理検証を行う場合が考えられる。

【００５６】この第三の実施の形態での論理シミュレータ８０２は機能レベル命令シミュレータであり、論理シミュレータ８０２をマイクロ命令動作として説明する。情報処理装置８０１（Ｄ）の上で、論理エミュレータ制御プログラム１０９と、処理制御プログラム２０１と、論理シミュレータ８０２を起動する。この３つのプログラムの間では、処理制御プログラム２０１の接続プラグインタフェース２０７によって論理シミュレータ８０２が接続され、プロセス間通信制御部２０３によってプロセス間通信を行う。よってシステム全体を統括し、コントロールしているのは処理制御プログラム２０１である。

【００５７】論理検証の手順としては、第一の実施の形態の要領で論理エミュレータ装置１０１を作動させ動作可能状態とする。次に論理シミュレータ８０２を処理制御プログラム２０１が起動し、論理シミュレータ８０２に与えるテストベクトル（テスト命令列）８０６をＲＡＭモデル８０５にセットする。なお、ここで取り上げる論理シミュレータ８０２は一般的な機能レベル命令論理シミュレータとし、処理制御部８０３と論理モデル８０４とＲＡＭモデル８０５を構成要素とするものとして説明を行う。

【００５８】次に論理シミュレータ８０２でセットされたテストベクトル８０６のテスト命令列を順次シミュレーションすると、特殊な意味を持つ命令列をシミュレーションするとき、その特殊命令列がマイクロ命令動作を伴う場合に、論理シミュレータ８０２は、インタフェース情報３０６に現在のＲＡＭモデル８０５の内容をセットし、マイクロ命令動作を論理エミュレータ装置１０１に対して要求する。この要求は処理制御部８０３経由で処理制御プログラム２０１に伝えられ、セットされたインタフェース情報を論理エミュレータ装置１０１のインタフェース情報３０６に転送する。論理シミュレータ８０２は要求を出した後、引き続きテスト命令列のシミュレーションを再開する。また、インタフェース情報を転送した後、処理制御プログラム２０１は、論理エミュレータ装置１０１を動作させ、この時点で論理シミュレータ８０２と論理エミュレータ装置１０１が完全に非同期に動作を開始する。次に一定サイクルだけエミュレーションを行うとマイクロ命令動作が完了し、第一の実施の形態と同様の手順で論理エミュレータ装置１０１は動作を中断する。この時、中断状態をインタフェース情報として論理シミュレータ８０２に転送するが、非同期に動作している論理シミュレータ８０２と同期を取るのが同期／非同期制御部２１７である。

【００５９】この同期／非同期制御部２１７は、論理シミュレータ８０２がマイクロ命令動作要求を出したときに論理シミュレータ８０２側で論理エミュレーション完了時刻をあらかじめ予想しているため、論理シミュレータ８０２がテスト命令列を一定命令数だけシミュレーションすると待ち合わせを行う場合と、一定命令数だけシミュレーションが完了していない場合には、割込み処理として論理シミュレータ８０２に報告する場合とを制御する。

【００６０】報告されたエミュレーション完了報告を処理制御部８０３が管理し、テスト命令列のシミュレーションを制御する。これら一連の処理を繰り返し行い、論理検証を行う。このように論理シミュレータ８０２と論

理エミュレータ装置１０１が混在する論理検証方式では、インタフェース情報を互いに転送しあい、検証状態の整合性をとりながら、検証途中では完全に非同期に動作し、効率の良い論理検証を行う。また、論理シミュレータ８０２を接続する場合には、接続プラグインタフェース２０７を用いて接続するため、接続インタフェースを統一することが可能であれば、どのような論理検証手法であっても接続が可能であり、更に複数の異なる論理検証手段を一括して取扱うことが可能である。

【００６１】以上のように第一の実施の形態で説明した通り、検証対象論理構造モデル１０５に組み込まれた擬似論理モデル３０４と、テストベクトル３０３ａに組み込まれたテスト制御プログラム３０５と、これらと情報の授受を行うことで、外部から論理エミュレーションを制御する処理制御プログラム２０１を組み込むことで、論理エミュレータ装置１０１の高速性を最大限に利用した高速、且つ、操作性や観測性に優れた対話型の論理検証方式の確立が可能となる。

【００６２】また、第二の実施の形態に例示したように、処理制御プログラム２０１が、たとえばＯＳ等の実行環境に依存しないＴｃｌ／Ｔｋ等の汎用言語で記述されていることにより、第一の実施の形態で使用したテストベクトル３０３や処理制御プログラム２０１をそのまま使用して、検証論理チップ５０８等の実チップの論理検証が可能な環境を構築することができ、論理モデルから実チップに至るまでの一貫した論理検証が可能となり、論理検証を効率良く行うことが可能である。

【００６３】また、第三の実施の形態に例示したように、論理シミュレータ８０２と論理エミュレータ装置１０１等のように、検証レベルの異なる論理検証手法を接続し、インタフェース情報を互いに転送することで非同期動作の整合性をとった論理検証が可能であり、たとえば、マイクロプロセッサとその周辺機器のＩ／Ｏ制御デバイス等で構成されるシステム全体等のような、大規模論理の装置全体としての一貫した論理検証が可能である。

【００６４】以上本発明者によってなされた発明を実施の形態に基づき具体的に説明したが、本発明は前記実施の形態に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。

【００６５】

【発明の効果】本発明の論理検証方法によれば、論理チップ全体での論理エミュレーションによるシステム論理検証を短時間に効率よく行うことができる、という効果が得られる。

【００６６】また、マンマシンインタフェースを充実させ、論理エミュレーションにおける内部状態の観測性や操作性、さらにはテスト容易性を向上させることができる、という効果が得られる。

【００６７】また、論理モデルから実チップに至るまでの一貫した論理検証を実現することができる、という効果が得られる。

【００６８】また、論理検証の環境構築に要する工数や期間を短縮して、論理検証工程におけるコスト削減を実現することができる、という効果が得られる。

【図面の簡単な説明】

【図１】本発明の第一の実施の形態である論理検証方法が実施される情報処理システムの構成の一例を示す概念図である。

【図２】本発明の論理検証方法を実現するためのソフトウェアの構成の一例を示した概念図である。

【図３】本発明の第一の実施の形態である論理検証方法の作用の一例を示すフローチャートである。

【図４】本発明の第二の実施の形態である論理検証方法が実施される情報処理システムの構成の一例を示す概念図である。

【図５】本発明の第二の実施の形態である論理検証方法にて用いられるソフトウェアの構成の一例を示した概念図である。

【図６】本発明の第二の実施の形態である論理検証方法の作用の一例を示すフローチャートである。

【図７】本発明の第三の実施の形態である論理検証方法にて用いられるソフトウェアの構成の一例を示した概念図である。

【図８】本発明の論理検証方法の参考技術である論理エミュレーションシステムの構成の一例を示す概念図である。

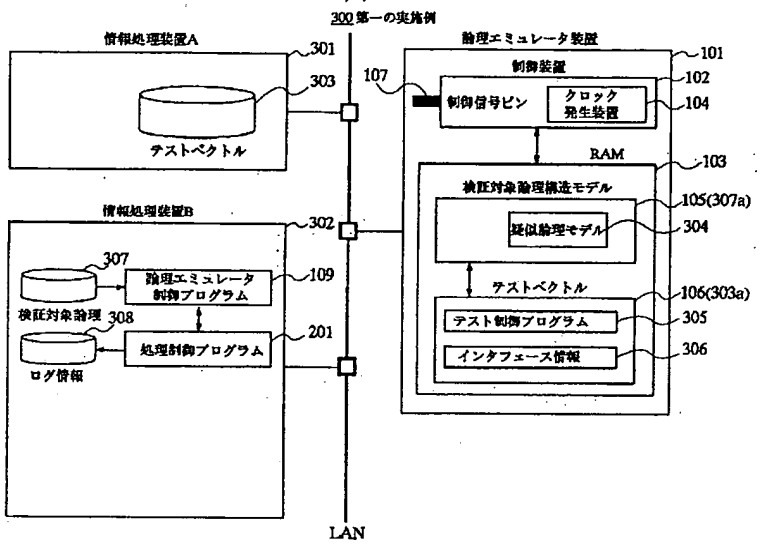
【符号の説明】

１００…論理エミュレーションシステム、１０１…論理エミュレータ装置、１０２…制御装置、１０３…ＲＡＭ、１０４…クロック発生装置、１０５…検証対象論理構造モデル格納エリア、１０６…テストベクトル格納エリア、１０７…制御信号ピン、１０８…情報処理装置、１０９…論理エミュレータ制御プログラム、１１０…制御部、１１１…論理コンパイラ、１１２…ＲＡＭアクセス部、１１３…ＧＵＩ制御部、１１３…グラフィカルユーザーインタフェース制御部、１１４…検証対象論理、１１４ａ…検証対象論理構造モデル、２０１…処理制御プログラム、２０２…ＧＵＩ制御部、２０３…プロセス間通信制御部、２０４…論理エミュレータ制御部、２０５…テストベクトル制御部、２０６…データ変換部、２０７…接続プラグインタフェース、２０８…トリガーイベント制御部、２０９…動作制御部、２１０…ＲＡＭアクセス部、２１１…処理制御部、２１２…表示部、２１３…入力制御部、２１４…ウインドウ制御部、２１５…設定情報制御部、２１６…ログデータ採取制御部、２１７…同期／非同期制御部、３００…論理エミュレーションシステム、３０１…情報処理装置、３０２…情報処理装置、３０３…テストベクトル、３０３ａ…テストベクトル、３０４…擬似論理モデル（第２の論理構造モデ

ル）、３０５…テスト制御プログラム、３０６…インタフェース情報、３０７…検証対象論理、３０７ａ…検証対象論理構造モデル（第１の論理構造モデル）、３０８…ログ情報、５００…論理エミュレーションシステム、５０１…情報処理装置、５０２…専用情報処理装置、５０３…Ｔｃｌ／Ｔｋインタプリタ、５０４…ログ情報、５０５…デバイスドライバ、５０６…パラレルＩ／Ｏインタフェース、５０７…制御装置、５０８…検証論理チ＊ップ（目的論理装置）、５０９…ＲＡＭ、６０１…ドライバ制御プログラム、６０２…制御部、６０３…処理制御プログラムインタフェース部、６０４…デバイスドライバ制御部、６０５…ログデータ採取制御部、８００…論理エミュレーションシステム、８０１…情報処理装置、８０２…論理シミュレータ、８０３…処理制御部、８０４…論理モデル、８０５…ＲＡＭモデル、８０６…テストベクトル。

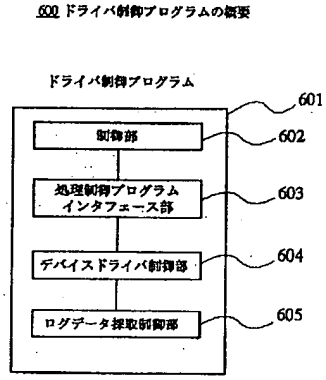
【図１】

図　1



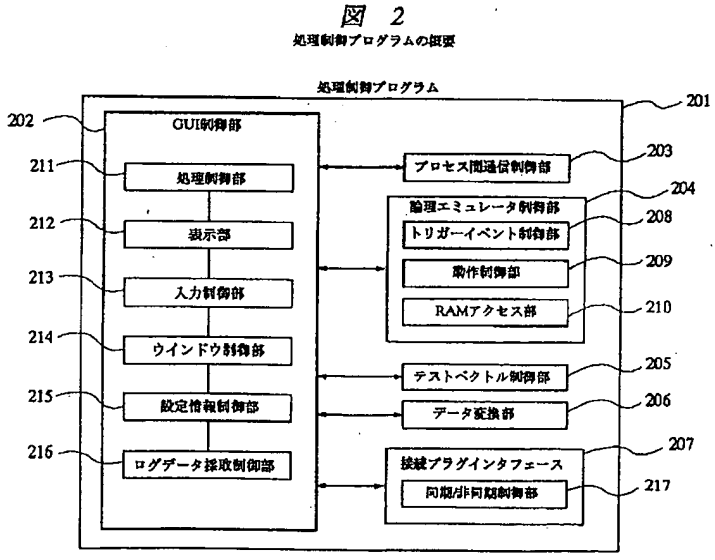
【図５】

図　5



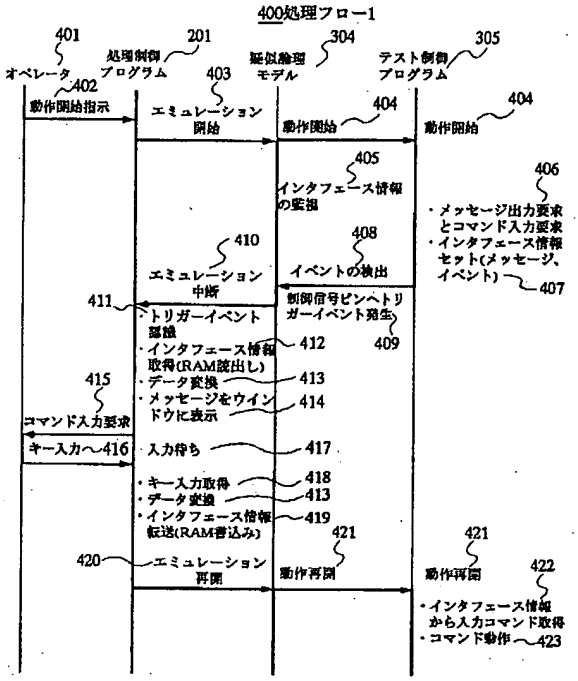
【図２】

図　2



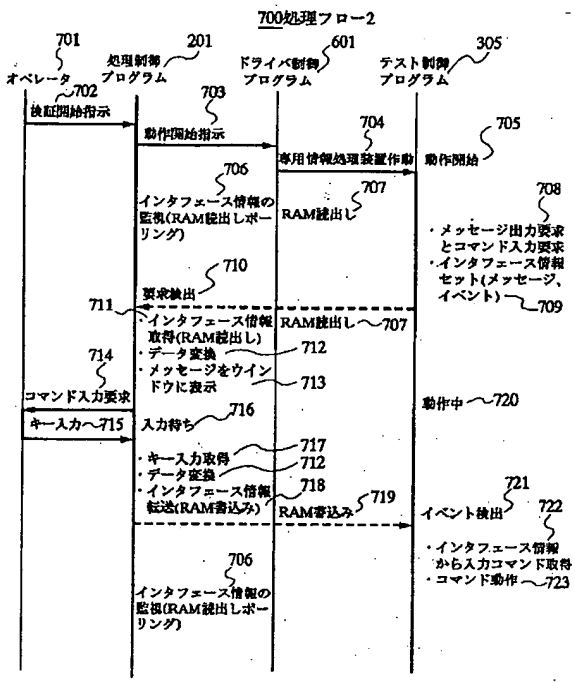
【図３】

図 3



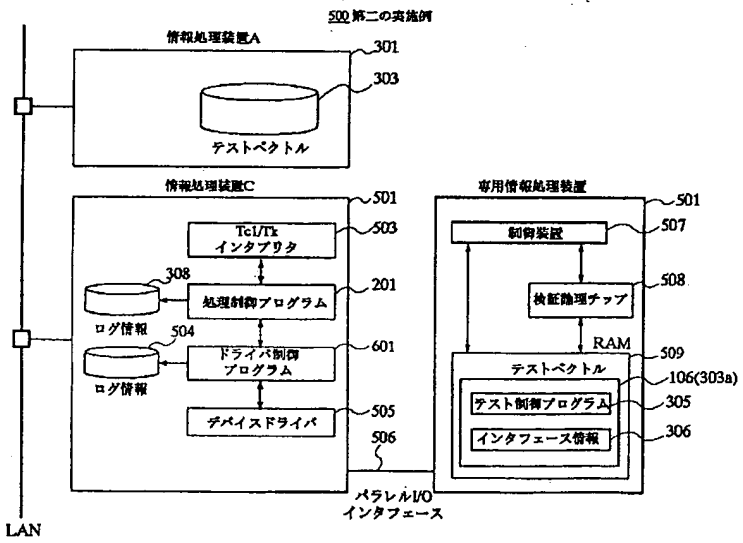
【図６】

図 6



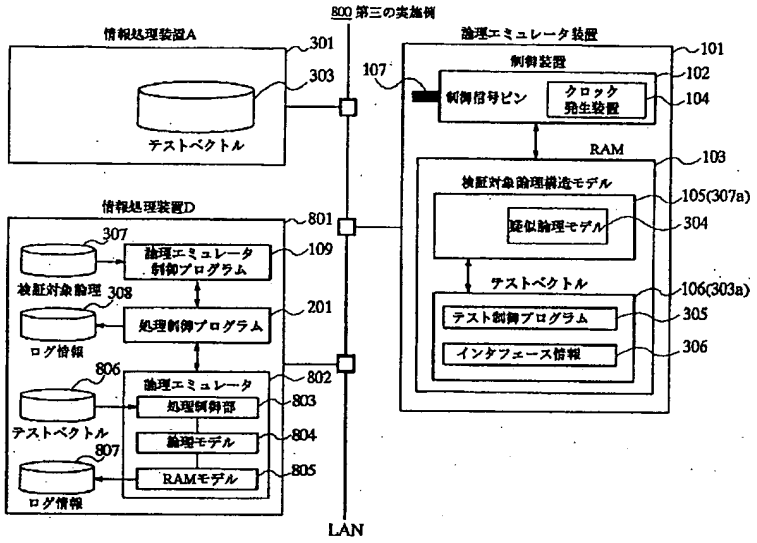
【図４】

図 4



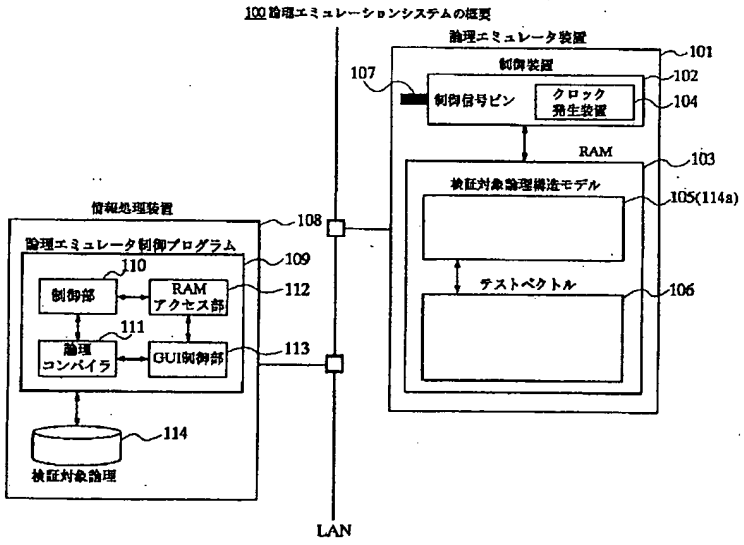
【図７】

図 7



【図８】

図 8



フロントページの続き

(51)Int.Cl.[7]	識別記号	F I	ﾃｰﾏｺｰﾄﾞ(参考)
		H01L 21/82	T

Ｆターム(参考) 2C032 AA01 AC08 AE07 AE08 AE10
5B046 AA08 BA03 JA05
5B048 AA01 BB02 DD01 DD05 DD15
5F064 HH05 HH09 HH10 HH13 HH14
9A001 BZ05 DZ13 HZ32 JJ49 JZ45